

# INDEX

S NO	Date	Title
1	21/1/25	Azure DevOps Environment SetUp
2	21/1/25	Azure DevOps Project SetUp and User Story Management
3	28/1/25	Setting Up epics, Feature and User Stories for Project Planning
4	11/2/25	Sprint Planning
5	18/2/25	Poker Estimation
6	25/2/25	Design the Class and Sequence Diagram
7	4/3/25	Design the Activity and Use Case Diagram
8	25/3/25	Testing: Test Plans and Test Case
9	15/4/25	Pipelines Creations
10	22/4/25	GitHub Project Structure

**EXP NO: 1**

## **AZURE DEVOPS ENVIRONMENT SETUP**

### **Aim:**

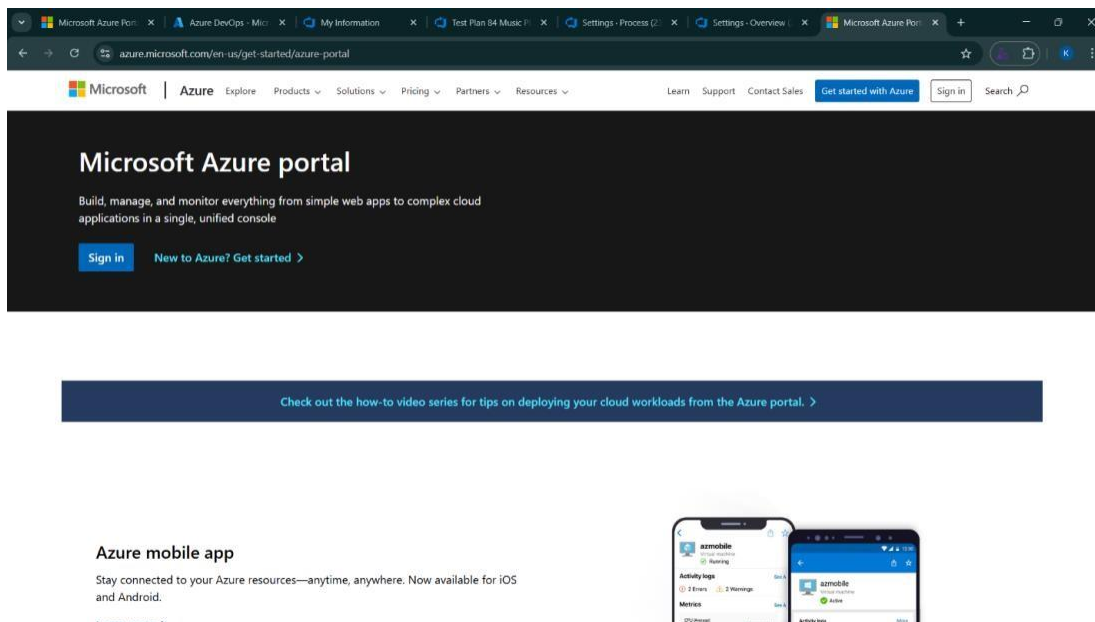
To set up and access the Azure DevOps environment by creating an organization through the Azure portal.

### **INSTALLATION**

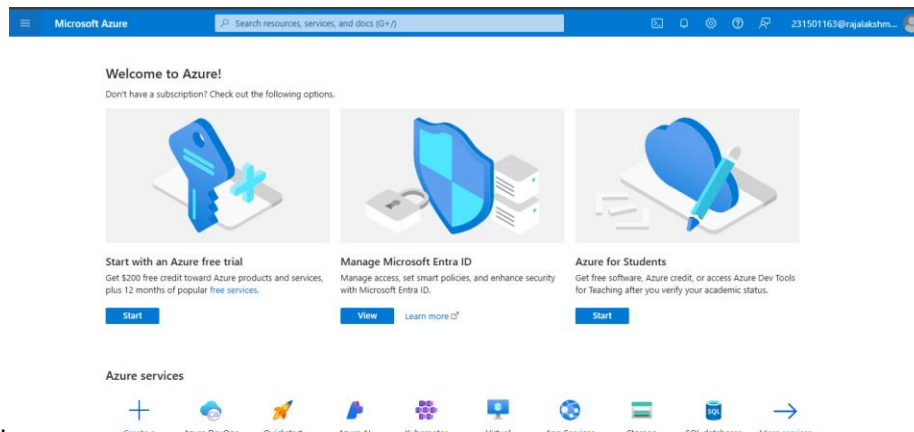
1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/get-started/azure-portal>.

Sign in using your Microsoft account credentials.

If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>

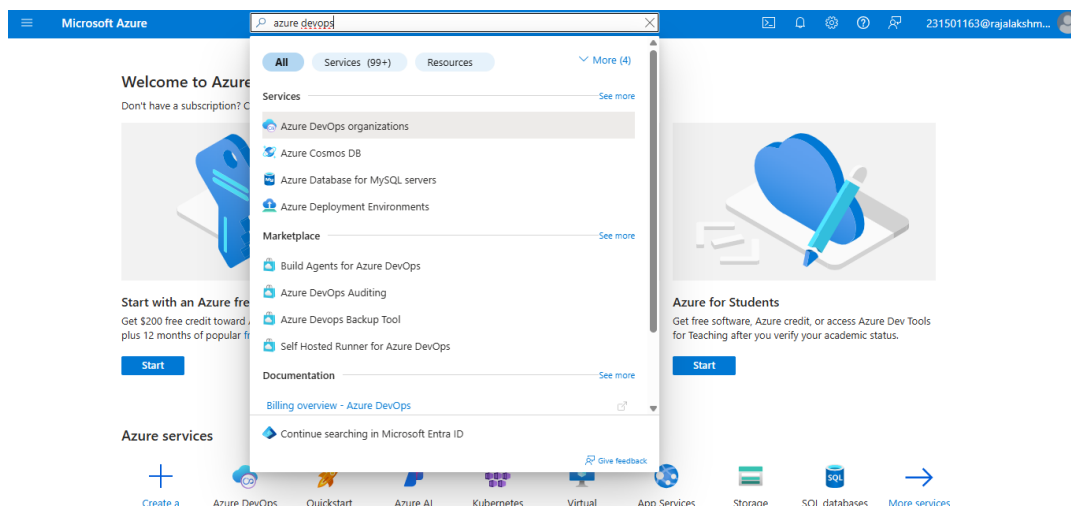


## 2. Azure home page



3.

4. Open DevOps environment in the Azure platform by typing *Azure DevOps Organizations* in the search bar.



5. Click on the *My Azure DevOps Organization* link and create an organization and you should be taken to the Azure DevOps Organization Home page.

Microsoft Azure Search resources, services, and docs (G+/)

Home > Azure DevOps


We've made it easier to manage Azure DevOps billing and subscriptions. You can [set up billing](#), [change your subscription](#) or pay for more users and resources within Azure DevOps. [Learn more](#)

## Azure DevOps

Plan smarter, collaborate better, and ship faster with a set of modern dev services

[My Azure DevOps Organizations](#)

[Get started using Azure DevOps](#)  
[Billing management for Azure DevOps](#)



**Result:**

Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

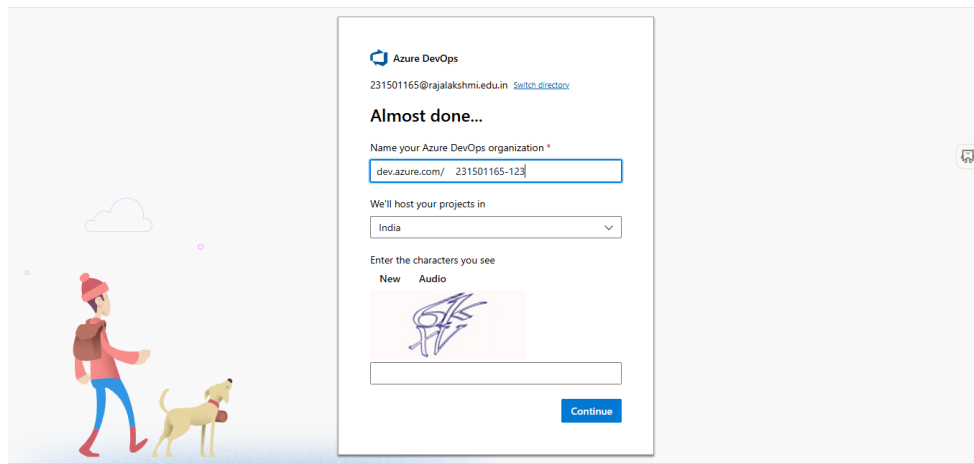
EXP NO: 2

## AZURE DEVOPS PROJECT SETUP AND USER STORY MANAGEMENT

**Aim:**

To set up an Azure DevOps project for efficient collaboration and agile work management.

1. Create an Azure Account



2. Create the First Project in Your Organization

a. After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.

b. On the organization's **Home page**, click on the **New Project** button.

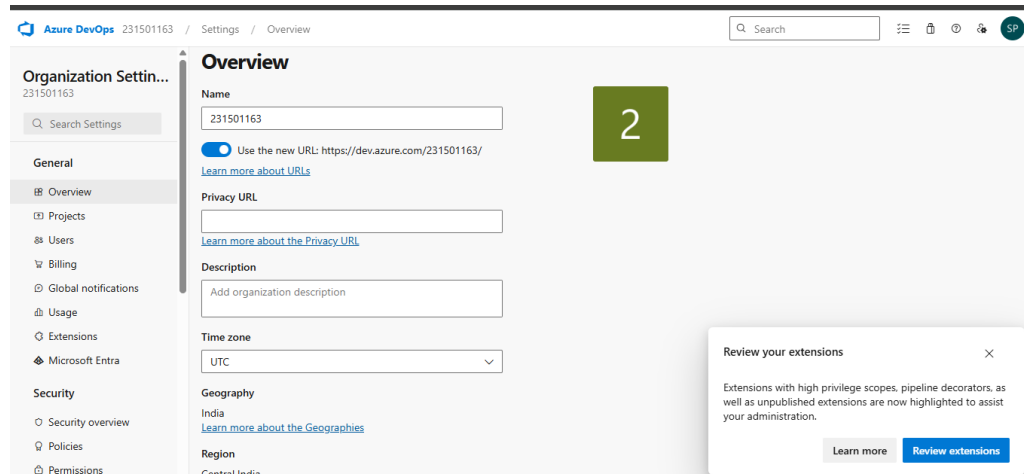
c. Enter the project name, description, and visibility options:

**Name:** Choose a name for the project (e.g., **LMS**).

**Description:** Optionally, add a description to provide more context about the project.

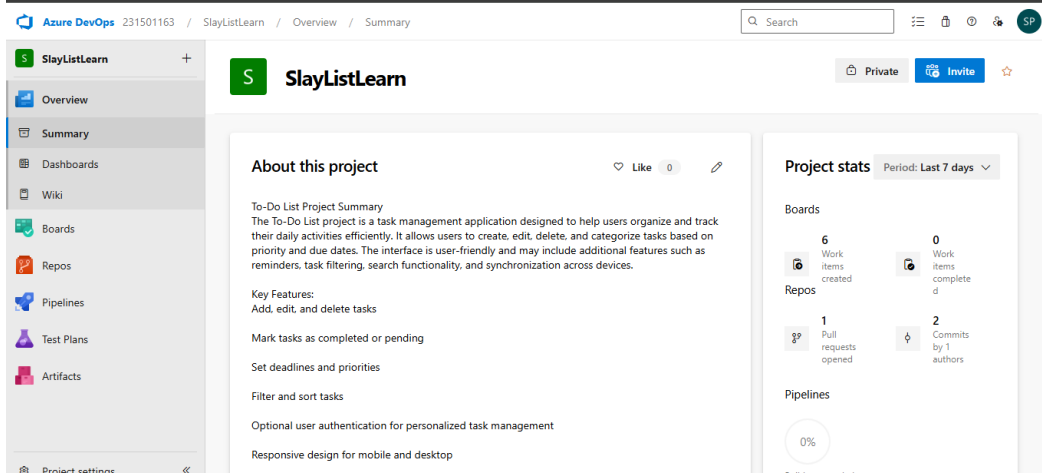
**Visibility:** Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).

d. Once you've filled out the details, click **Create** to set up your first project.



3. Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to the Azure DevOps Organization Home page.

## 4. Project dashboard

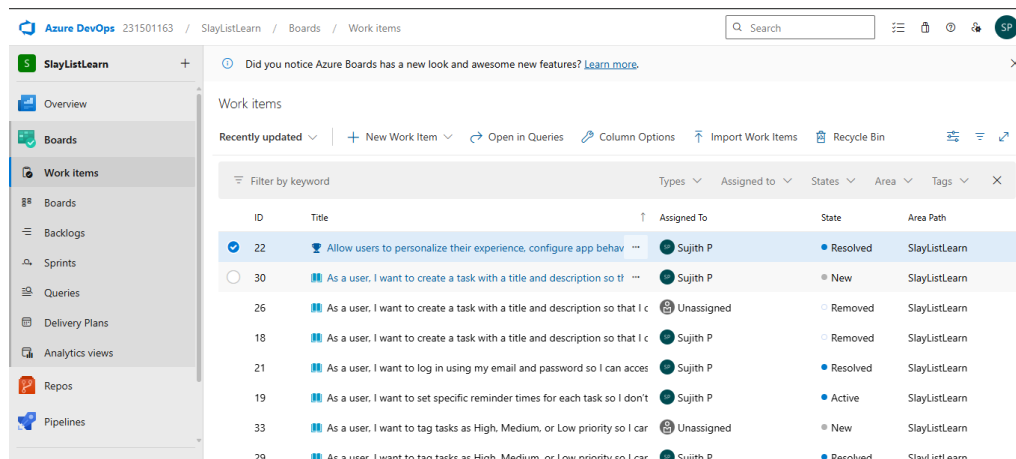


## 5. To manage user stories:

a. From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards** page, where you can manage work items, backlogs, and sprints.

b. On the **work items** page, you'll see the option to **Add a work item** at the top. Alternatively, you can find a + button or **Add New Work Item** depending on the view you're in. From the **Add a work item** dropdown, select **User Story**. This will open a form to enter details for the new User Story.

c.



Azure DevOps 231501163 / SlayListLearn / Boards / Work items

Did you notice Azure Boards has a new look and awesome new features? [Learn more.](#)

Work items

Recently updated ▾ | + New Work Item ▾ | → Open in Queries | Column Options | Import Work Item

Filter by keyword | Types ▾ | Assigned to ▾

ID	Title	Assigned To	State	Category
22	Allow users to personalize their experience, configure app behav	Sujith P	Resolved	SlayListLearn
30	As a user, I want to create a task with a title and description so that I c	Sujith P	New	SlayListLearn
26	As a user, I want to create a task with a title and description so that I c	Unassigned	Removed	SlayListLearn
18	As a user, I want to create a task with a title and description so that I c	Sujith P	Removed	SlayListLearn
21	As a user, I want to log in using my email and password so I can acces	Sujith P	Resolved	SlayListLearn
19	As a user, I want to set specific reminder times for each task so I don't	Sujith P	Active	SlayListLearn
33	As a user, I want to tag tasks as High, Medium, or Low priority so I car	Unassigned	New	SlayListLearn
29	As a user, I want to tag tasks as High, Medium, or Low priority so I car	Sujith P	Resolved	SlayListLearn

Microsoft Sign out

Sujith P  
231501163@rajalakshmi.edu.in  
[My Microsoft account](#)  
[Switch directory](#)

Sign in with a different account

**Results:**

Successfully created an Azure DevOps project with user story management and agile workflow setup.



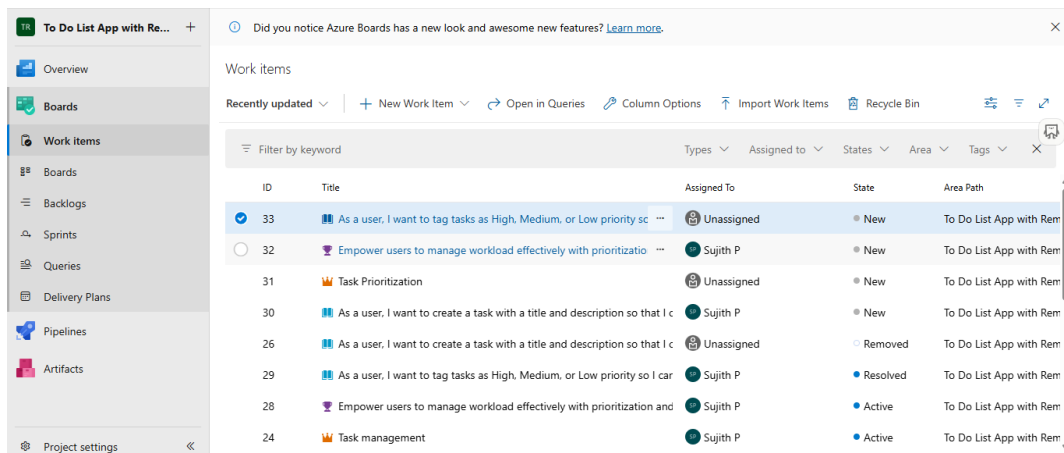
EXP NO: 3

## SETTING UP EPICS, FEATURES, AND USER STORIES FOR PROJECT PLANNING

### Aim:

To learn about how to create epics, user story, features, backlogs for your assigned project.

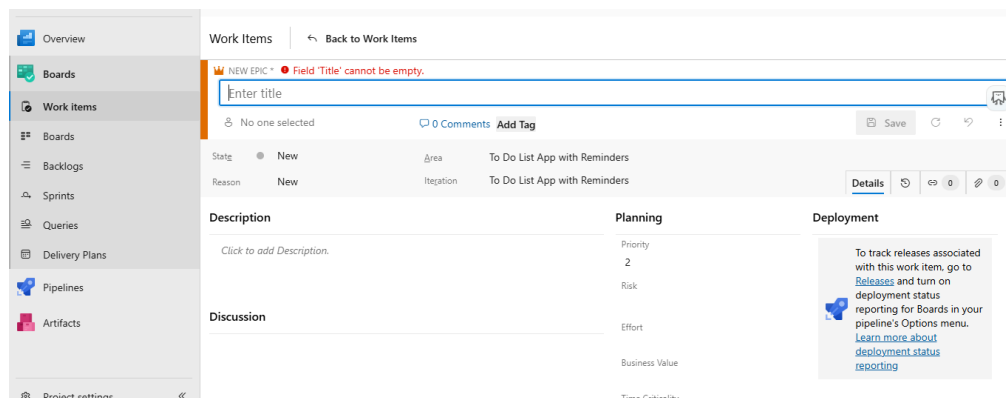
### Create Epic, Features, User Stories, Task



The screenshot shows the Azure Boards interface. On the left is a sidebar with navigation options: Overview, Boards, Work Items (selected), Backlogs, Sprints, Queries, Delivery Plans, Pipelines, and Artifacts. The main area displays a list of work items under the heading 'Work items'. Above the list are filters and actions: 'Recently updated', '+ New Work Item', 'Open in Queries', 'Column Options', 'Import Work Items', and 'Recycle Bin'. The list has columns for ID, Title, Assigned To, State, and Area Path. The items are sorted by 'Recently updated'.

ID	Title	Assigned To	State	Area Path
33	As a user, I want to tag tasks as High, Medium, or Low priority sc	Unassigned	New	To Do List App with Rem
32	Empower users to manage workload effectively with prioritization	Sujith P	New	To Do List App with Rem
31	Task Prioritization	Unassigned	New	To Do List App with Rem
30	As a user, I want to create a task with a title and description so that I c	Sujith P	New	To Do List App with Rem
26	As a user, I want to create a task with a title and description so that I c	Unassigned	Removed	To Do List App with Rem
29	As a user, I want to tag tasks as High, Medium, or Low priority so I car	Sujith P	Resolved	To Do List App with Rem
28	Empower users to manage workload effectively with prioritization and	Sujith P	Active	To Do List App with Rem
24	Task management	Sujith P	Active	To Do List App with Rem

### 1. Fill in Epics



The screenshot shows the 'NEW EPIC' form in Azure Boards. The sidebar is the same as in the previous screenshot. The main area has a header 'Work Items' with a 'Back to Work Items' link. Below the header is a red error message: 'Field "Title" cannot be empty.' There is a text input field for the title. Below the input field are buttons for 'No one selected', '0 Comments', and 'Add Tag'. There are also 'Save', 'Cancel', and 'Close' buttons. The form is divided into three sections: 'Description', 'Planning', and 'Deployment'. The 'Description' section has a 'Click to add Description.' placeholder. The 'Planning' section has fields for 'Priority' (set to 2), 'Risk', 'Effort', 'Business Value', and 'Time Estimation'. The 'Deployment' section has a text area with a placeholder: 'To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting'.

## 2.Fill in Features

The screenshot shows the 'New Feature' form in Azure DevOps. The left sidebar contains navigation links: Overview, Boards, Work Items, Backlogs, Sprints, Queries, Delivery Plans, Pipelines, and Artifacts. The main area is titled 'Work Items' and includes a 'Back to Work Items' link. A message at the top states: 'NEW FEATURE - Field "Title" cannot be empty.' Below this is a text input field for the title. The form is divided into three main sections: Description, Planning, and Deployment. The Description section has a 'Click to add Description.' link. The Planning section includes fields for Priority (set to 2), Risk, Effort, Business Value, and Time Criticality. The Deployment section has a 'Click to add Deployment.' link and a note about tracking releases. A 'Save' button is located at the top right of the form.

## 3.Fill in User Story Details

The screenshot shows the 'New User Story' form in Azure DevOps. The left sidebar is the same as in the previous screenshot. The main area is titled 'Work Items' and includes a 'Back to Work Items' link. A message at the top states: 'NEW USER STORY - Field "Title" cannot be empty.' Below this is a text input field for the title. The form is divided into three main sections: Description, Planning, and Deployment. The Description section has a 'Click to add Description.' link. The Planning section includes fields for Story Points, Priority (set to 2), Risk, and Classification. The Deployment section has a 'Click to add Deployment.' link and a note about tracking releases. A 'Save' button is located at the top right of the form.

### Result:

Thus, the creation of epics, features, user story and task has been created successfully.

EXP NO: 4

# SPRINT PLANNING

## Aim:

To assign user story to specific sprint for the TO DO List Project.

## Sprint Planning:

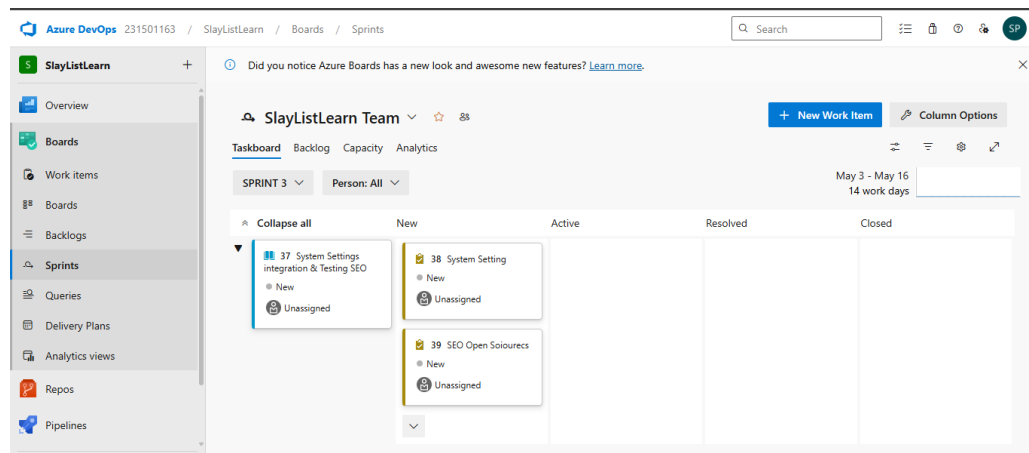
### Sprint 1:

Order	Title	State	Assigned To	Rem...
1	As a user, I want to set specific reminder times for each task...	Active	Sujith P	
2	As a user, I want to tag tasks as High, Medium, or Low priority...	Resolved	Sujith P	
3	As a user, I want to log in using my email and password so I...	Resolved	Sujith P	
4	As a user, I want to change my password and manage sessi...	Closed	Sujith P	
5	As a user, I want to tag tasks as High, Medium, or Low priority...	Resolved	Sujith P	
6	As a user, I want to create a task with a title and description...	New	Sujith P	
7	As a user, I want to tag tasks as High, Medium, or Low priority...	New		

### Sprint 2

Task ID	Title	State	Assigned To
19	As a user, I want to set specific reminder times for each task so I don't forget important deadlines.	Active	Sujith P
20	As a user, I want to tag tasks as High, Medium, or Low priority so I can focus on what's most important.	Resolved	

## Sprint 3



### Result:

The Sprints are created for the To Do List Project.

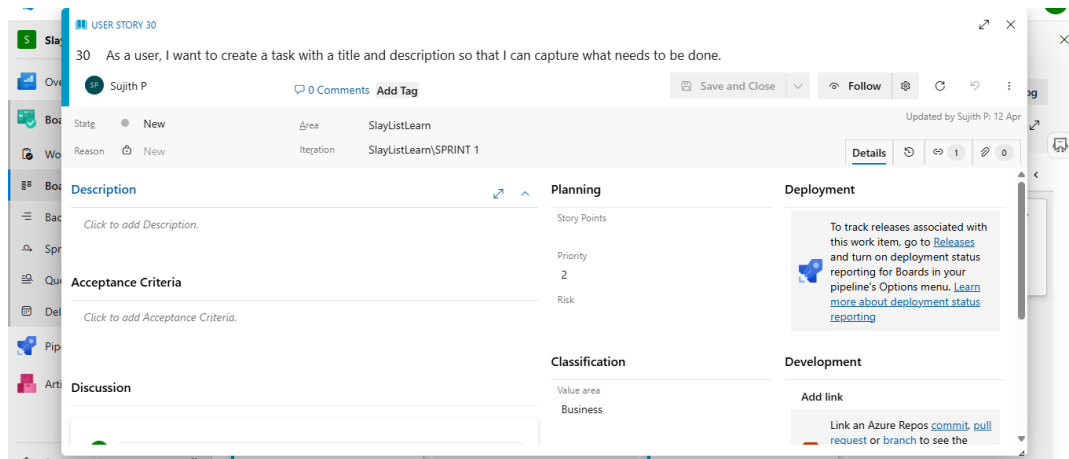
**EXP NO: 5**

# POKER ESTIMATION

**Aim:**

Create Poker Estimation for the user stories – TO DO List Project.

**Poker Estimation**



**Result:**

The Estimation/Story Points is created for the project using Poker Estimation.

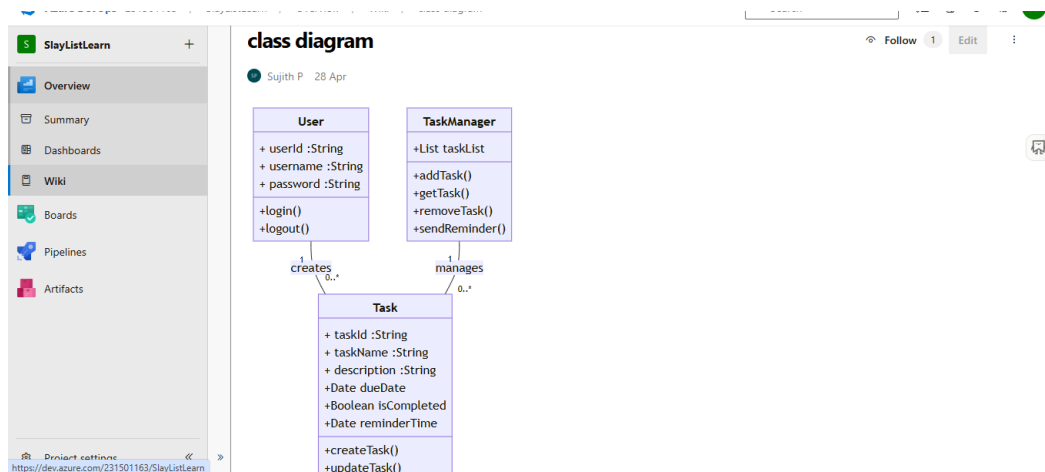
**EXP NO: 6**

## **DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE**

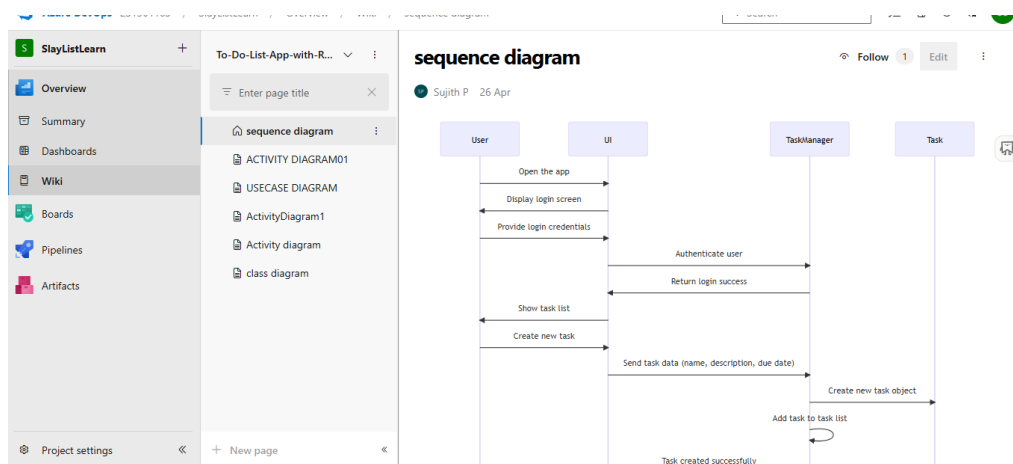
### **Aim:**

To Design a Class Diagram and Sequence Diagram for the given Project.

### **6A. Class Diagram**



### **6B. Sequence Diagram**



### **Result:**

The Class Diagram and Sequence Diagram is designed Successfully for the TO DO List.

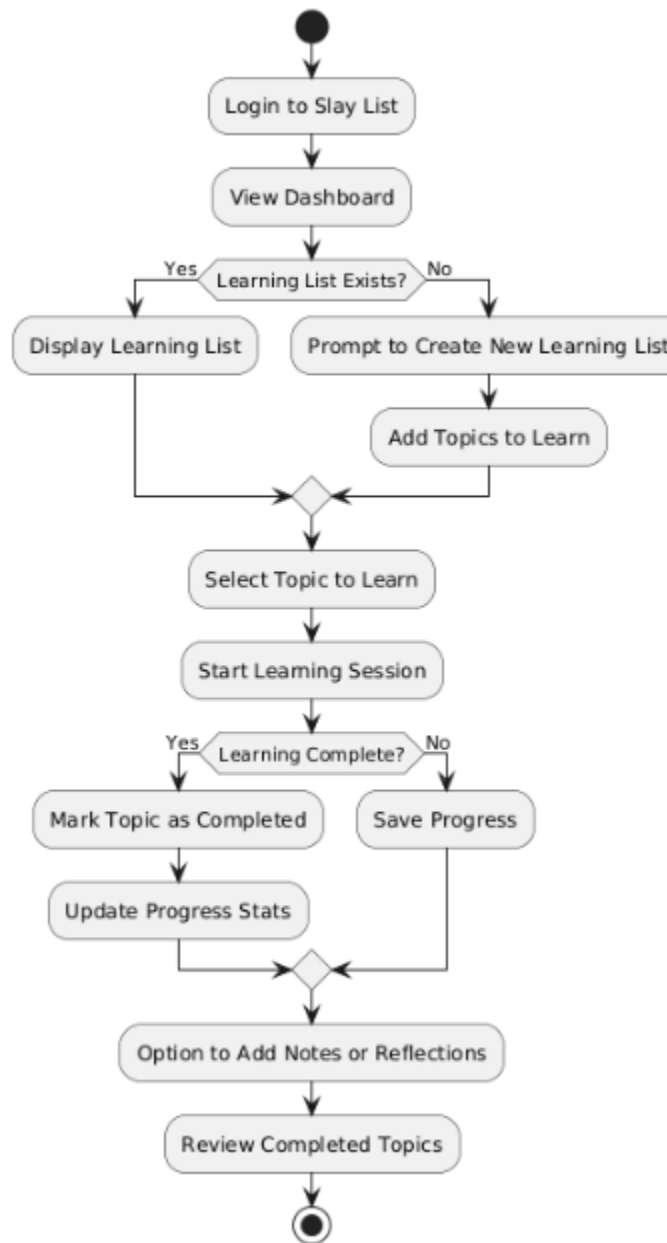
**EXP NO: 7**

## **DESIGNING ACTIVITY AND USE CASE DIAGRAMS FOR PROJECT ARCHITECTURE**

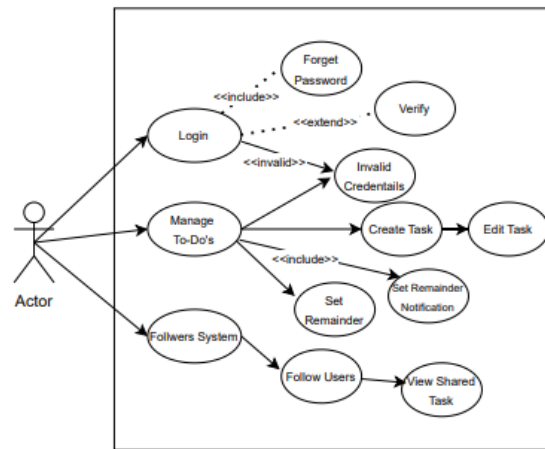
**Aim:**

To Design an Activity and Use Case Diagram for the given Project.

### **7A. Activity Diagram**



## 7B. Use Case Diagram



### Result:

The Activity and the Use Case Diagram for the given Project is successfully executed.



**EXP NO: 8**

## **TESTING – TEST PLANS AND TEST CASES**

**Aim:**

Test Plans and Test Case and write two test cases for at least four user stories showcasing the happy path and error scenarios in azure DevOps platform.

### **Test Planning and Test Case**

#### **Test Case Design Procedure**

##### **1. Understand Core Features of the Application**

- User Signup & Login
- Viewing and Managing Playlists
- Fetching Real-time Metadata
- Editing playlists (rename, reorder, record)
- Creating smart audio playlists based on categories (mood, genre, artist, etc.)

##### **2. Define User Interactions**

- Each test case simulates a real user behavior (e.g., logging in, renaming a playlist, adding a song).

##### **3. Design Happy Path Test Cases**

- Focused on validating that all features function as expected under normal conditions.
- Example: User logs in successfully, adds item to playlist, or creates a category-based playlist.

##### **4. Design Error Path Test Cases**

- Simulate negative or unexpected scenarios to test robustness and error handling.
- Example: Login fails with invalid credentials, save fails when offline, no recommendations found.

##### **5. Break Down Steps and Expected Results**

- Each test case contains step-by-step actions and a corresponding expected outcome.
- Ensures clarity for both testers and automation scripts.

##### **6. Use Clear Naming and IDs**

- Test cases are named clearly (e.g., TC01 – Successful Login, TC10 – Save Playlist Fails).
- Helps in quick identification and linking to user stories or features.

##### **7. Separate Test Suites**

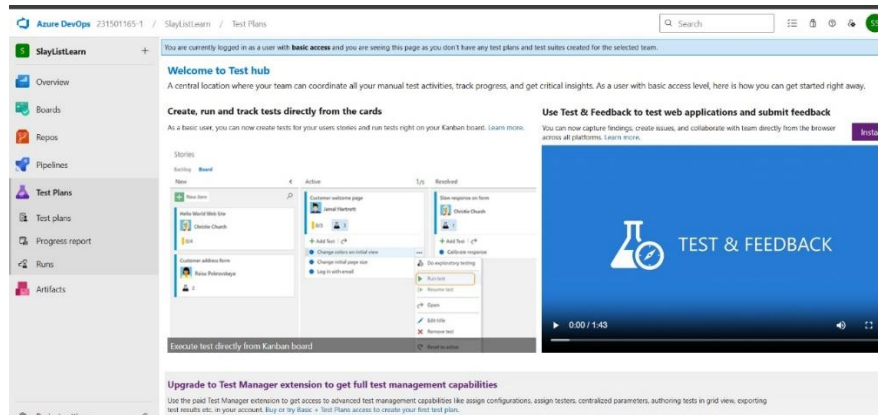
- Grouped test cases based on functionality (e.g., Login, Playlist Editing, Recommendation System).

- Improves organization and test execution flow in Azure DevOps.

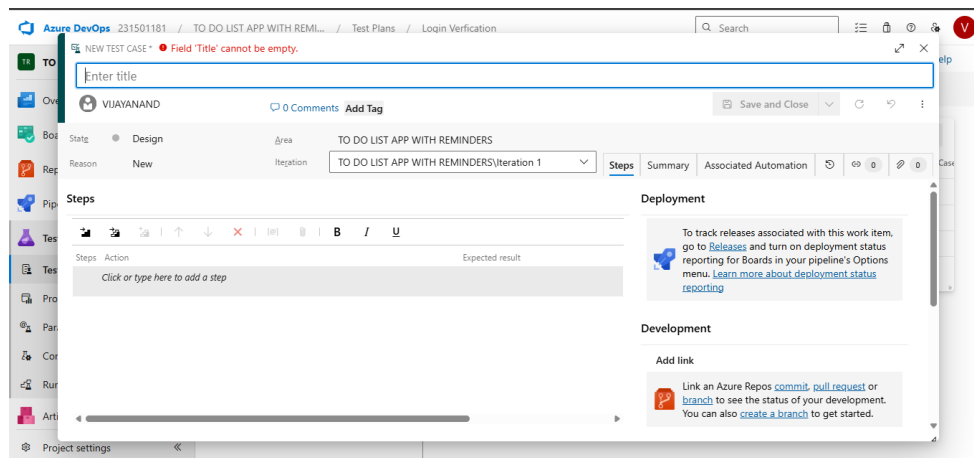
## 8. Prioritize and Review

- Critical user actions are marked high-priority.
- Reviewed for completeness and traceability against feature requirements.

## 1.New test plan



## 2. Test suite



### 3. Test case

Give two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

Music Playlist Batch Creator – Test Plans

#### USER STORIES

- As a user, I want to sign up and log in securely so that I can access my playlists (ID: 79).
- As a user, I need to see my playlist in one place (ID: 76).
- As a user, I should be able to create an audio playlist as needed (ID: 73).
- As a user, I should be able to rename, record, and change the playlist (ID: 68).
- As a user, I need to have real-time metadata (ID: 65).

#### Test Suites

##### Test Suit: TS01 - User Login (ID: 86)

###### 1. TC01 – Successful Sign Up

- **Action:**
  - Go to the Sign-Up page.
  - Enter valid name, email, and password.
  - Click "Sign Up".
- **Expected Results:**
  - Sign-Up form is displayed.
  - Fields accept values without error.
  - Account is created, and the user is redirected to the dashboard.
- **Type:** Happy Path

###### 2. TC02 – Secure Login

- **Action:**
  - Go to the Login page.
  - Enter valid email and password.
  - Click on "Login".
- **Expected Results:**
  - Login form is displayed.
  - Fields accept data without error.
  - User is logged in and redirected to the dashboard.
- **Type:** Happy Path

###### 3. TC03 – Sign Up with Existing Email

- **Action:**
  - Go to the Sign-Up page.
  - Enter a name and an already registered email.
  - Click on "Sign Up".
- **Expected Results:**

- Fields accept data.
- Error message "Email already registered" is displayed.
- **Type:** Error Path

#### 4. TC04 – Login with Wrong Password

- **Action:**
  - Go to the Login page.
  - Enter valid email and incorrect password.
  - Click on "Login".
- **Expected Results:**
  - Input is accepted.
  - Error message "Invalid username or password" is shown.
- **Type:** Error Path

### Test Suit: TS02 – Add List (ID: 87)

#### 1. TC05 – View TO DO List Page

- **Action:**
  - Log in successfully.
  - Navigate to "My " section.
- **Expected Results:**
  - All created playlists are displayed clearly.
- **Type:** Happy Path

#### 2. TC06 –List Failure

- **Action:**
  - Disconnect from the internet.
  - Navigate to "List".
- **Expected Results:**
  - Network is offline.
  - Error message "Unable to load playlists" is shown.
- **Type:** Error Path

### Test Suit: TS03 – Real Time TO DO Socket (ID: 88)

#### 1. TC07 – Real-Time Metadata Display

- **Action:**
  - Display the list panel.
  - Observe the metadata panel.
- **Expected Results:**
  - Metadata (title, user, list, duration) is displayed and updates in real time.
- **Type:** Happy Path

#### 2. TC08 – Metadata Not Updating

- **Action:**

- Add different attribute.
  - Observe the metadata panel.
- **Expected Results:**
  - Metadata remains static or shows default/fallback message.
- **Type:** Error Path

#### **Test Suit: TS04 - Editing (ID: 89)**

##### **1. TC09 – Rename Edit Successfully**

- **Action:**
  - Navigate to “Rename the List ”.
  - Click "Rename" next to a playlist.
  - Enter a new name and click "Save".
- **Expected Results:**
  - List details / name updates successfully.
- **Type:** Happy Path

##### **2. TC10 – Rename with Blank Name**

- **Action:**
  - Click "Rename" on a playlist.
  - Leave the field blank.
  - Click "Save".
- **Expected Results:**
  - Error message "List name cannot be empty" is shown.
- **Type:** Error Path

##### **3. TC11 – Change Priority Order**

- **Action:**
  - Open a List.
  - Drag and drop list to reorder.
  - Click "Save".
- **Expected Results:**
  - Playlist order is updated and saved.
- **Type:** Happy Path

##### **4. TC12 – Change Priority Order Fails**

- **Action:**
  - Login and go to “My Playlists”.
  - Select a List.
  - Go offline or simulate server error.
  - Reorder songs and click “Save Order”.
- **Expected Results:**
  - Error message: "Failed to update order. Please check your connection".
- **Type:** Error Path

## Test Cases

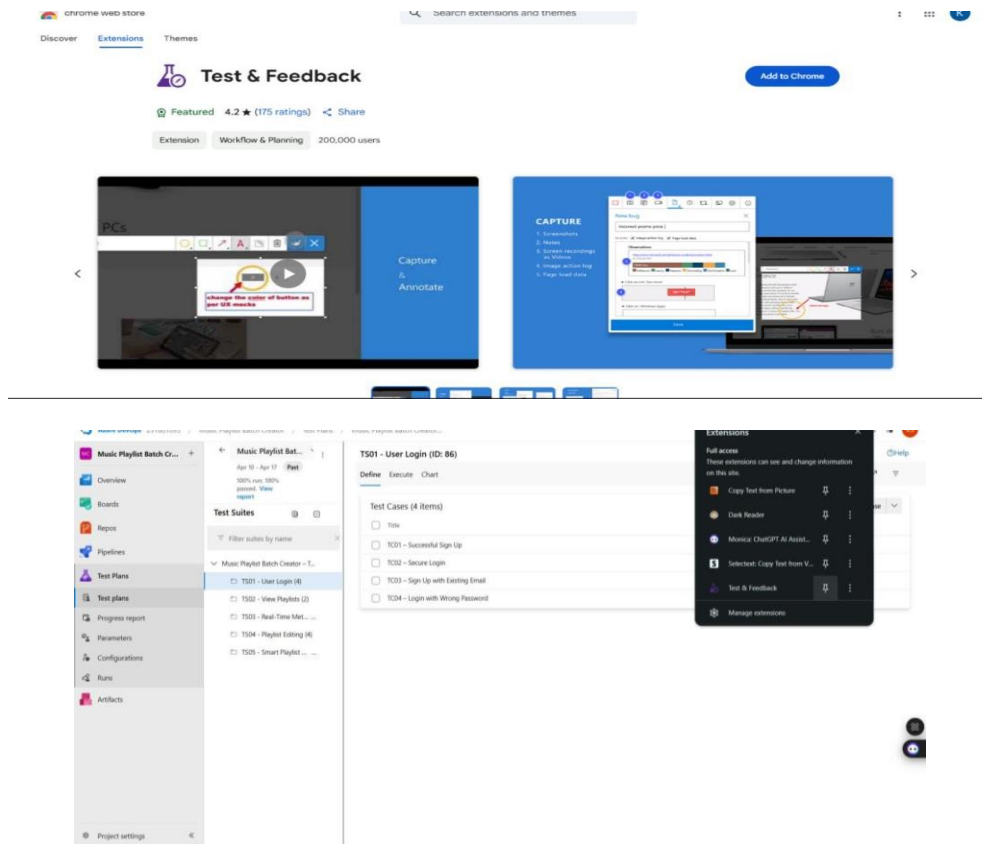
The screenshot shows the 'Test Suites' section of a testing tool. The left sidebar contains a navigation menu with items: Overview, Boards, Repos, Pipelines, Test Plans, Test plans (selected), Progress report, Parameters, Configurations, Runs, and Artifacts. The main area displays the 'Login Verification (ID: 26)' suite. It shows a status bar with 'May 15 - May 22', 'Current', and '100% run, 100% passed. View report'. Below this is a 'Test Suites' list with a filter 'Filter suites by name' and a single entry 'Login Verification (4)'. The right pane shows the 'Test Points (4 items)' table.

Title	Outcome	
Login Verification of correct User	Passed	1
LoginVerification for Wrong Info	Passed	2
Add the TO DO Item to the list	Passed	3
Application Setting Completion	Passed	4

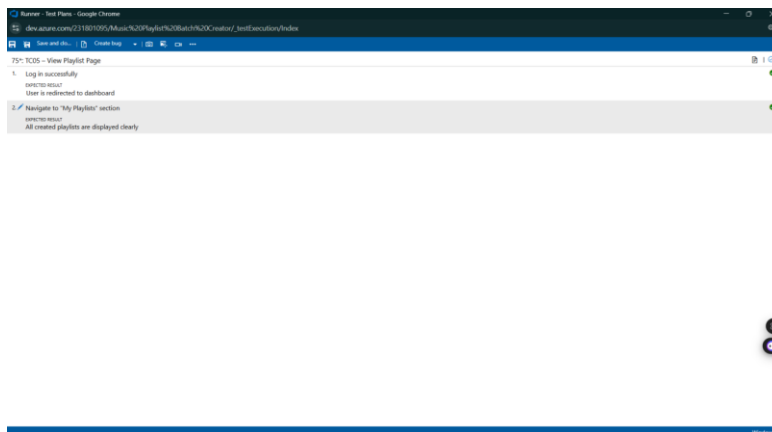
The screenshot shows the details of a test case titled '27\*: Login Verification of correct User'. It lists three steps, each with a status icon (green checkmark) and a magnifying glass icon. The first step is 'Open the browser' with an expected result of 'Redirect to the Main SlayListLearn page'. The second step is 'Click the log in Button'. The third step is 'Enter the correct details'.

1. Open the browser  
EXPECTED RESULT  
Redirect to the Main SlayListLearn page
2. Click the log in Button
3. Enter the correct details

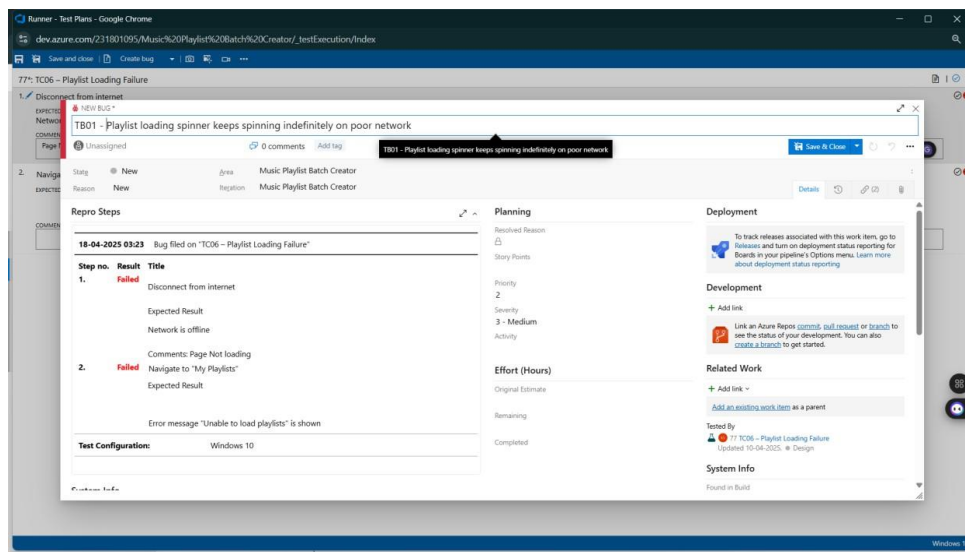
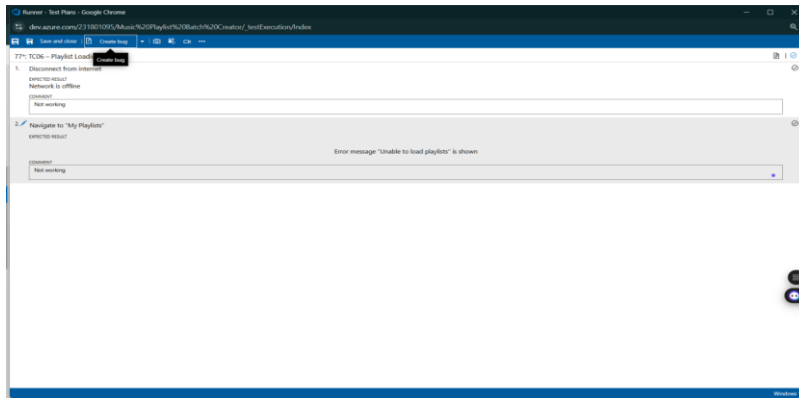
## 4. Installation of test



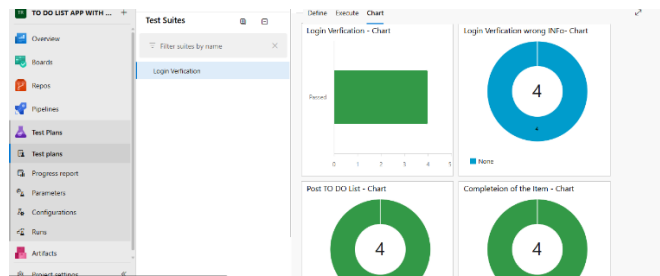
## 5. Running the test cases



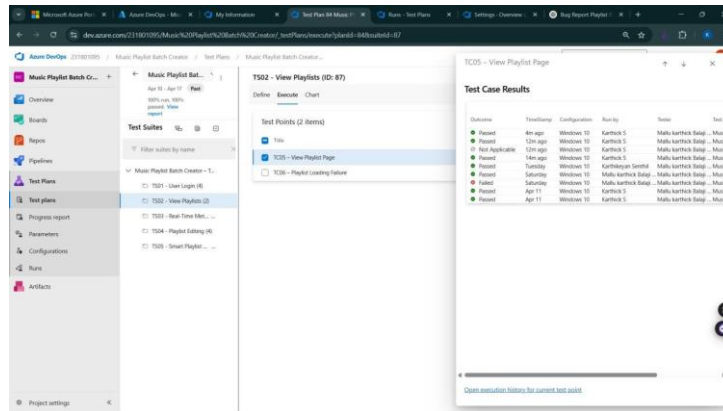
## 6. Creating the bug



## Test Case Results:







## Result:

The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path

**EXP NO: 9**

## **PIPE LINES IN AZURE DEVOP'S**

**Aim:**

To create an Azure Pipeline resource and run the performance and integrate the targeted endpoint.

**Load Testing**

**Steps to Create an Azure Load Testing Resource:**

Before you run your first test, you need to create the Azure Load Testing resource:

1. **Sign in to Azure DevOps Portal**  
Go to <https://dev.azure.com> and log in with your Microsoft account.
2. **Navigate to Your Project**
3. Select your organization and project
4. If you haven't created a project yet, click "**New Project**", fill in the details, and create one.
5. **Create a New Pipeline**
6. On the left sidebar, go to **Pipelines**.
7. Click "**New Pipeline**" at the top right.
8. **Select Your Repository**
9. Choose where your code is stored (e.g., Azure Repos Git, GitHub, Bitbucket, etc.).
10. Authenticate and select the specific repository you want to build.
11. **Configure the Pipeline**
12. Choose a pipeline configuration method (YAML or classic editor).
  - a. **YAML:** Recommended for modern DevOps workflows.
  - b. **Classic editor:** Useful if you prefer a GUI-based configuration.
13. **Set Up Your Build Pipeline**
14. If using YAML:
  - a. Either let Azure generate a template based on your code, or write your own `azure-pipelines.yml` file.
15. If using classic:
  - a. Add tasks such as build, test, and deploy using the step-by-step editor.
16. **Save and Run the Pipeline**
17. Click "**Save and run**" to execute your pipeline for the first time.
18. Review the configuration and click "**Run**".
19. **Monitor the Pipeline Execution**
20. You'll be redirected to the run summary where you can see logs and results for each job and step.
21. *(Optional)* **Add Triggers and Variables**
22. Set up CI/CD triggers, environment variables, and other configurations to fine-tune your pipeline behavior.

## Creating Pipelines:

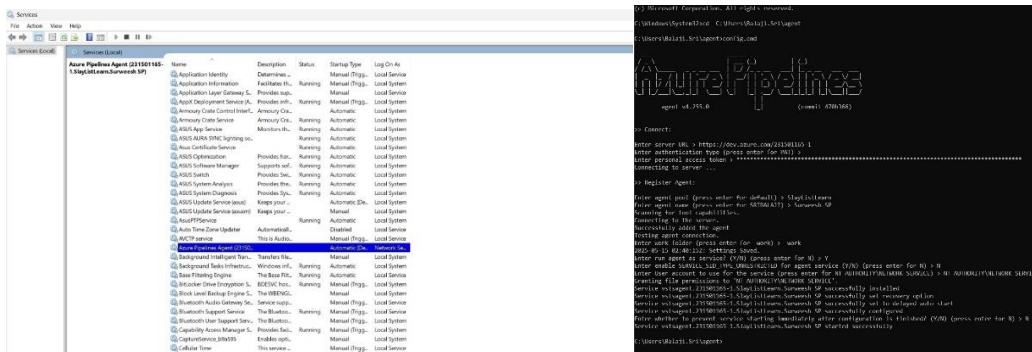
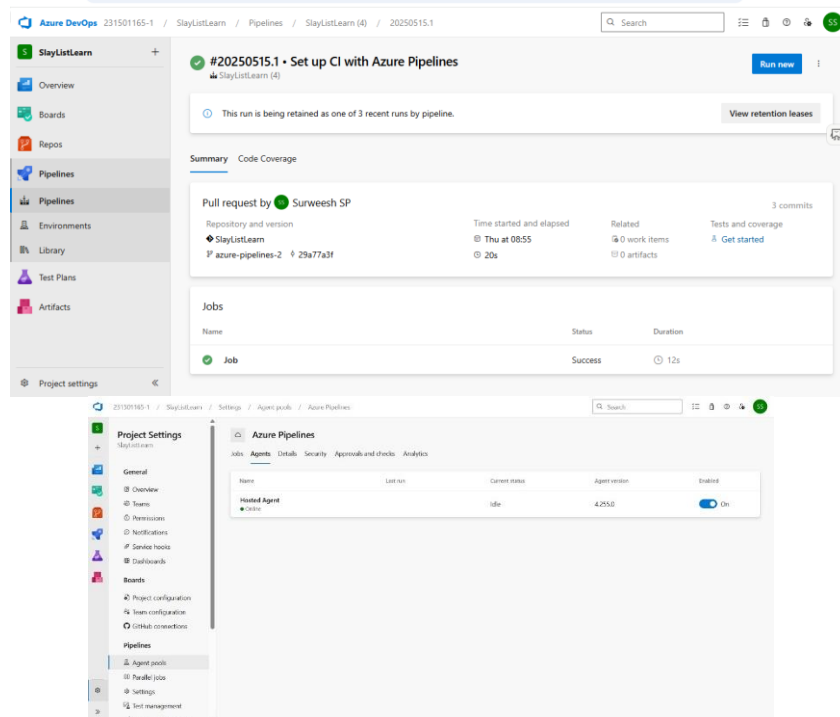
This screenshot shows the 'Recently run pipelines' table in the Azure DevOps interface. The table lists several pipelines, including 'SlayListLearn (6)', 'SlayListLearn (3)', 'SlayListLearn (4)', 'SlayListLearn (3)', 'Surweesh-SPOofboardProject', and 'SlayListLearn'. Each row displays the pipeline name, its last run details (including a trigger icon, a status icon, and a timestamp), and the last run time.

Pipeline	Last run
SlayListLearn (6)	#20250518.1 - Update azure-pipelines-1.yml for Azure Pipelines Trigger: Manually triggered for 'P' azure-pipelines Status: Success Time: Yesterday <1s
SlayListLearn (3)	#20250515.1 - Set up CI with Azure Pipelines Trigger: PR automated for 'P' azure-pipelines-1 Status: Success Time: Thursday <1s
SlayListLearn (4)	#20250515.1 - Set up CI with Azure Pipelines Trigger: PR automated for 'P' azure-pipelines-2 Status: Success Time: Thursday 25s
SlayListLearn (3)	#20250515.1 - Set up CI with Azure Pipelines Trigger: Individual CI for 'P' azure-pipelines Status: Success Time: Thursday <1s
Surweesh-SPOofboardProject	#20250514.2 - Set up CI with Azure Pipelines Trigger: Manually triggered for 'P' azure-pipelines Status: Success Time: Wednesday <1s
SlayListLearn	#20250514.1 - Set up CI with Azure Pipelines Trigger: Manually triggered for 'P' azure-pipelines Status: Success Time: Wednesday <1s

The first screenshot shows the 'Where is your code?' step in the 'New pipeline' wizard. It lists several options for where the code is stored: Azure Repos Git (YAML), Bitbucket Cloud (YAML), GitHub (YAML), and GitHub Enterprise Server (YAML). The second screenshot shows the 'Select a repository' step, where the user has selected 'SlayListLearn' from the list of repositories.

The third screenshot shows the 'Configure your pipeline' step, where the user has selected 'Starter pipeline' (Start with a minimal pipeline that you can customize to build and deploy your code). The fourth screenshot shows the 'Review your pipeline YAML' step, where the user is reviewing the generated YAML file for the 'Starter pipeline'. The YAML file is named 'azure-pipelines-1.yml' and contains the following content:

```
1 # Starter pipeline
2 # Start with a minimal pipeline that you can customize to build and deploy your code.
3 # Add steps that build, run tests, deploy, and more:
4 # https://aka.ms/yaml
5
6 trigger:
7   - main
8
9 pool:
10  vmImage: ubuntu-latest
11
12 steps:
13   - script: echo Hello, world!
14     displayName: 'Run a one-line script'
15
16   - script: |
17     echo Add other tasks to build, test, and deploy your project.
18     echo See https://aka.ms/yaml
19     displayName: 'Run a multi-line script'
```



## Result:

Successfully created the Azure Pipelines resource and executed a performance of the specified endpoint.

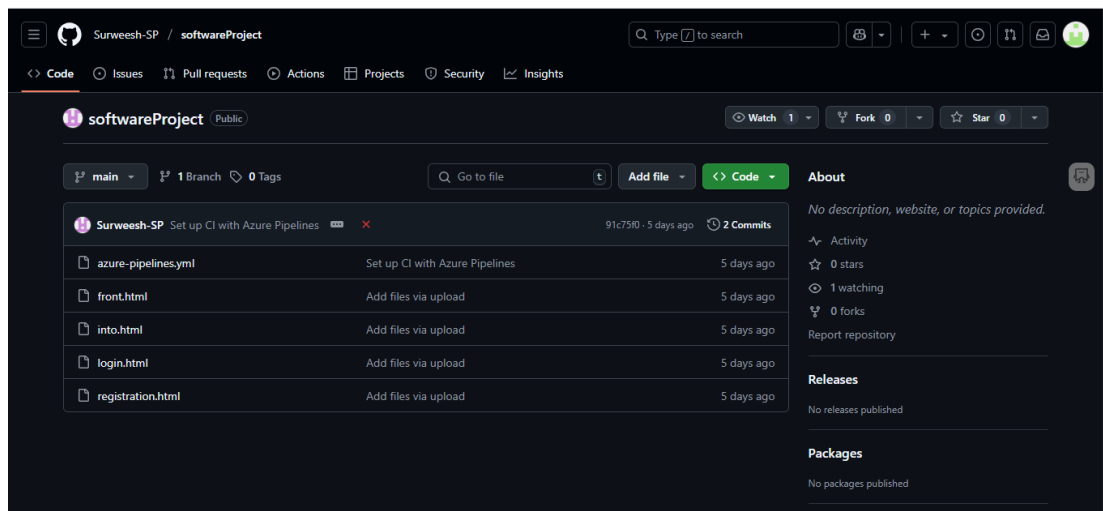
**EXP NO: 10**

## **GITHUB: PROJECT STRUCTURE & NAMING CONVENTIONS**

### **Aim:**

To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the TO DO List project.

### **GitHub Project Structure**



### **Result:**

The GitHub repository clearly displays the organized project structure and consistent naming conventions, making it easy for users and contributors to understand and navigate the codebase.